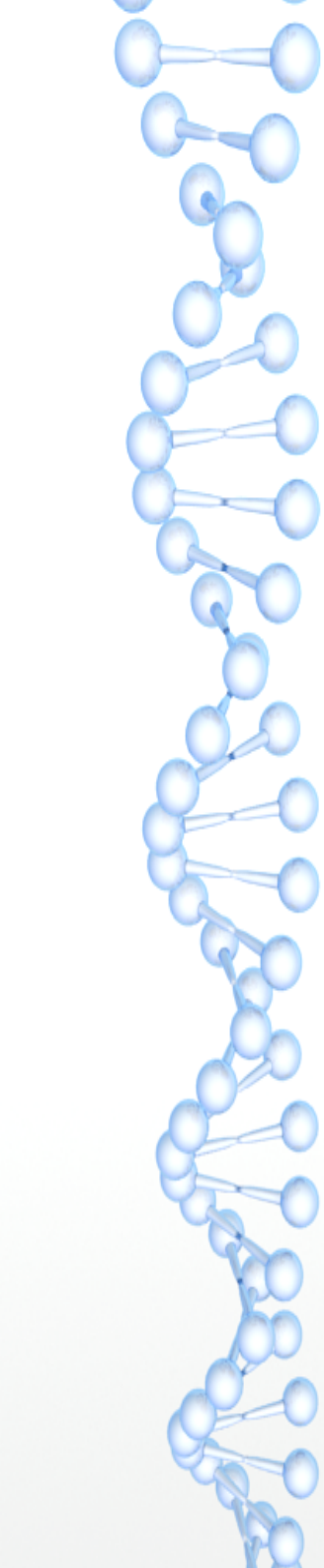
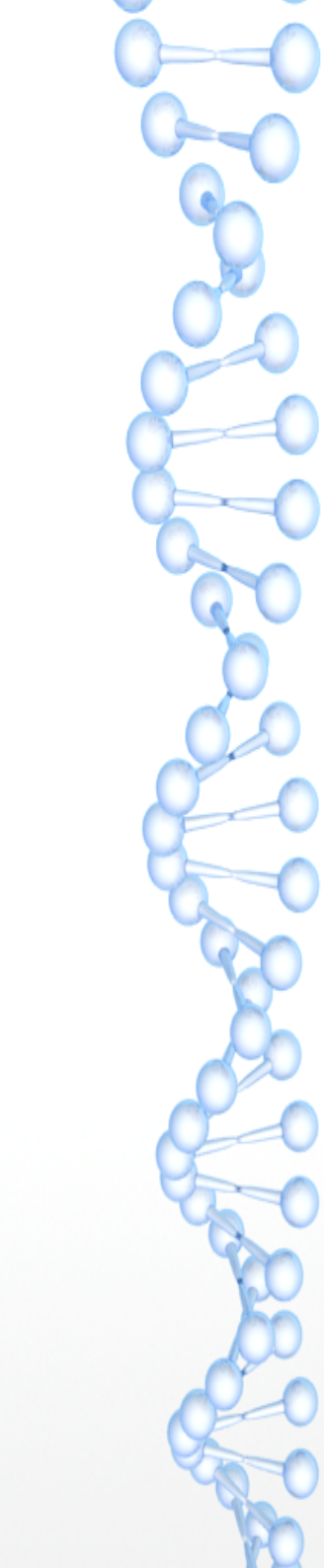


Introduction
Aux Systèmes multi-agents

Syma – cursus SCIA

Julien Saunier - julien.saunier@ifsttar.fr

IFSTTAR



Ressources

Multiagent Systems, A Modern Approach to Distributed Artificial Intelligence Edited by Gerhard Weiss, MIT Press.

emmanuel.adam.free.fr/site/IMG/pdf/poa_introP.pdf

www.limsi.fr/~jps/enseignement/

www.emse.fr/~boissier/

www.ecs.soton.ac.uk/about/pdfs/AgentLink_Fifty_Facts.pdf

www.agentlink.org/roadmap/al3rm.pdf

www.lirmm.fr/~ferber/

Cours librement inspiré de ceux de Flavien Balbo et Nicolas Sabouret

Les Systèmes Multi-Agents

Modèles d'interaction

Plan

- Rappels : Interactions indirectes
- La métaphore du tableau noir
- Les évolutions
- Les communications directes
  - Agent Communication Languages
  - Protocoles d'interaction
- Les communications multi-parties

Définitions : Interaction

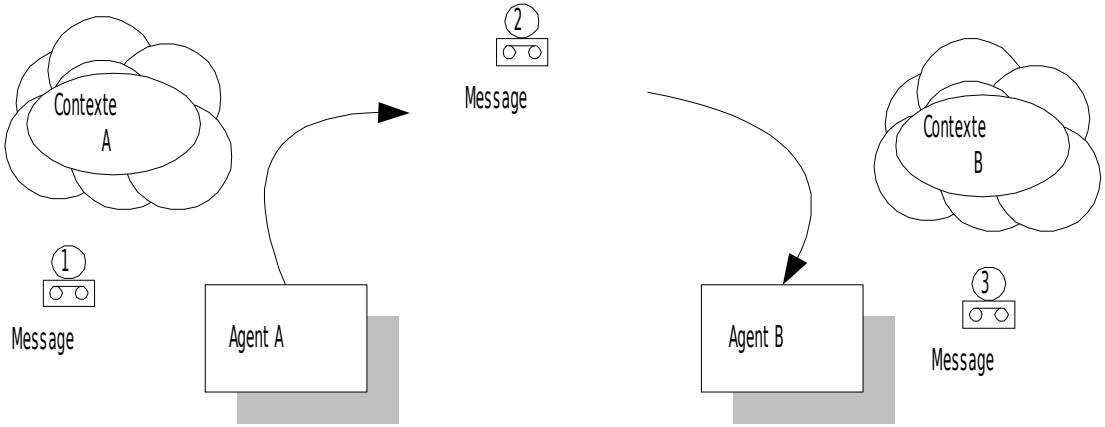
- Générale [Morin 77]
  - *Les interactions sont des actions réciproques modifiant le comportement ou la nature des éléments, corps, objets, phénomènes en présence ou en influence.*
- Multi-Agent
  - *Mise en relation dynamique de deux ou plusieurs agents par le biais d'actions réciproques.*

Définitions

- Quatre façons d'interagir entre agents [Werner89] :
  - *Pas de communication* (!= pas d'interaction): les agents ne communiquent pas, soit ils interagissent par perception de l'environnement, soit ils peuvent atteindre leur objectif sans aide extérieure.
  - *L'envoi de signaux* : les agents se synchronisent par l'envoi de messages codés.
  - *L'envoi de plans* : le transfert d'informations concerne les tâches et les croyances des agents.
  - *L'envoi de messages* : ce mode de communication, le plus usité dans la communauté SMA, permet aux agents d'échanger leurs intentions et leurs besoins.

Communication

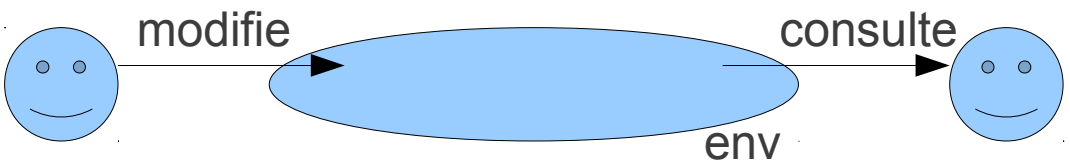
- **Problématique** [Shannon 48]



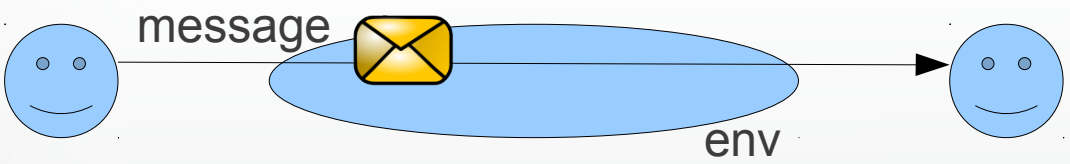
- **Définition** : La communication est l'échange volontaire d'informations provoqué par la production et la perception de symboles tirés d'un système partagé de symboles conventionnels [Russel et Norvig 03]

Résumé: interactions M.A.

- Avec l'environnement
  - Synchrone et bloquant → invocations !
- Avec les autres agents
  - Via l'environnement → interaction indirecte



    - Modèle « blackboard »
    - Réparti + interface → artefacts (Ricci & Omicini)
  - Par envoi de message → interaction directe



    - Support = environnement !

Interactions indirectes

- Mémoire partagée (blackboard)
  - → accès concurrent bloquant
    - N'empêche ni l'autonomie, ni les croyances
    - Simple à implémenter, pas de pb de concurrence
    - Ressource critique centralisée & bloquante !
- Stigmergie
  - → blackboard à N variables
    - perception locale (réduit blocage)
    - dynamique (diffusion, simulation fourmis)
- Tuple-spaces & Artefacts
  - → Artefact = objet dynamique avec interface

    mi-agent, mi-environnement (stigmergie)

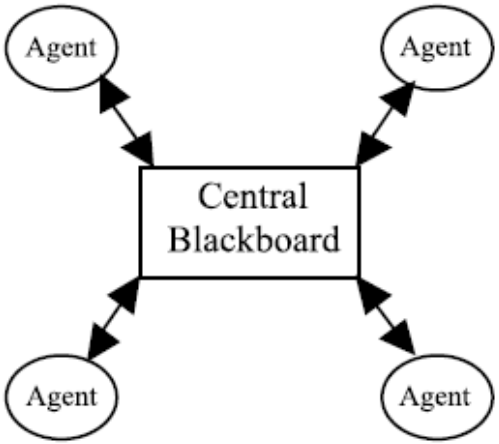
Interaction indirecte

Principe

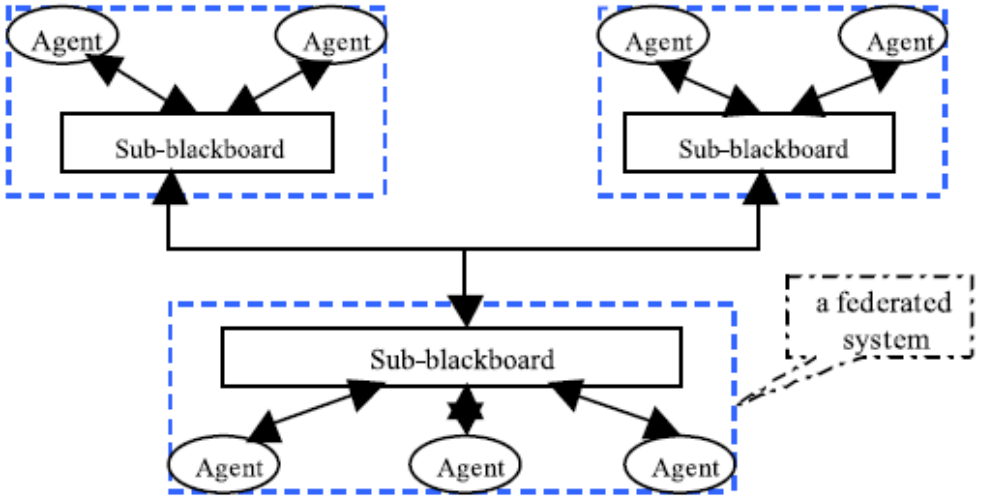
- Approche proposant d'ajouter au niveau de l'infrastructure un environnement logique pour faciliter les échanges d'information.
- Interaction dirigée par le « récepteur »
- Modalité :
  - Stigmergie,
  - Espace partagé : Blackboard, espace de tuples.
- Problématique:
  - Prise en compte du temps,
  - Gestion de l’ouverture / Problème de connexion,
  - Interprétation.

Interaction indirecte
Blackboard

- **Principe** [Engelmore 88] : Le tableau noir est un espace de recherche partagé où s'inscrivent les résultats obtenus par les sources de connaissances (KS / Agent). L'accès au tableau noir peut être en lecture et/ou écriture.



Central Blackboard Architecture



Distributed Blackboard Architecture

Interaction indirecte
Espace des tuples

- Origine : Système distribué.
- Principe [Carriero 86] : Le modèle Linda propose une mémoire partagée nommée espace de tuples, ainsi qu'un mécanisme de récupération des données utilisant leurs signatures.
- Mise en œuvre :
  - Un tuple est défini comme une suite ordonnée d'éléments typés.
  - Un motif (template) est un tuple particulier dans lequel les champs sont typés, mais peuvent ne pas avoir de valeur définie.
  - Trois opérateurs :
    - *out(t)* : ajout du tuple t,
    - *in(m)* : lecture d'un tuple correspondant au motif m et retrait de ce tuple,
    - *read(m)* : lecture d'un tuple correspondant au motif m, pas de modification de l'espace de tuples.

Agents & Artefacts

- Environnement = ensemble d'artefacts
- Artefact
  - Etat observable
  - Opérations lecture/écriture (non blocantes)
  - Opérateurs internes (environnement actif)
  - « Mode d'emploi »
    - Description du service
    - Description des opérations (interface)
- → Outil de coordination
- Implémentation : modèle CARTAGO (Ricci)

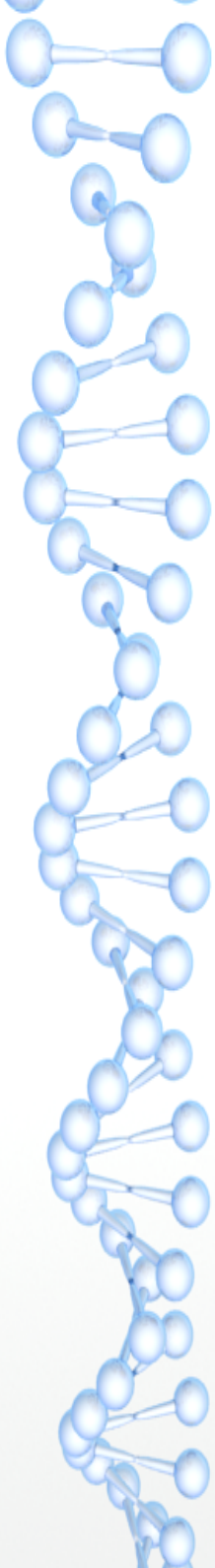
A&A (suite)

- Construction
  - createArtifact(Name,Type,Conf,Description)
  - getArtifactID(Name,Description):AID
  - disposeArtifact(AID)
- Utilisation
  - use(AID,Op(Name,Params),SID)
  - sense(SID,Filter,Timeout):Perception (timeout)
    - → L'artefact « informe » l'agent du résultat
  - focus(AID,SID) / unfocus
    - → L'agent « surveille » une donnée (SID)
  - getOI(AID):OI
    - → L'agent demande le mode d'emploi

Conclusion

(sur les interactions indirectes)

- Question du rôle de l'environnement
  - environnement ≠ agent
- Direct *vs* indirect → complémentaire !
  - Attention aux interblocages...



Interactions directes
Qui? Quoi? Comment?

- Théorie des actes de langage
- Modèle FIPA
- AUML

Interaction directe

Principe

- Approche classique de communication adressée : un émetteur envoie un message à un récepteur localisé par son adresse.
- Interaction dirigée par l'émetteur
- Modalités
  - Point à point, diffusion, diffusion restreinte.
- Problématique:
  - Prise en compte du temps,
  - Problème de connexion,
  - Interprétation,

Interaction directe

Qui ? Problème de connexion

- Problématique : Avec quel agent interagir pour obtenir un service, une ressource … ?

- Solutions :
  - Gestion des connaissances sociales.
  - Protocoles.
  - Service offert par la plateforme.

Interaction directe

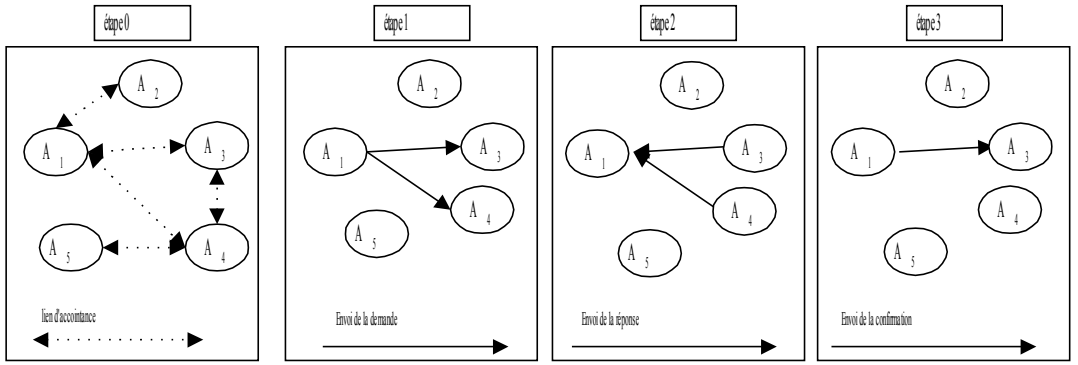
Diffusion

- Principe : les messages sont envoyés à tous les agents
- Avantages
  - simplicité,
  - prise en compte des besoins de l'émetteur et du récepteur.
  - gestion de l'ouverture.
- Limites
  - Coût bande passante,
  - Coût traitement,
  - Prise en compte du contexte.

Interaction directe

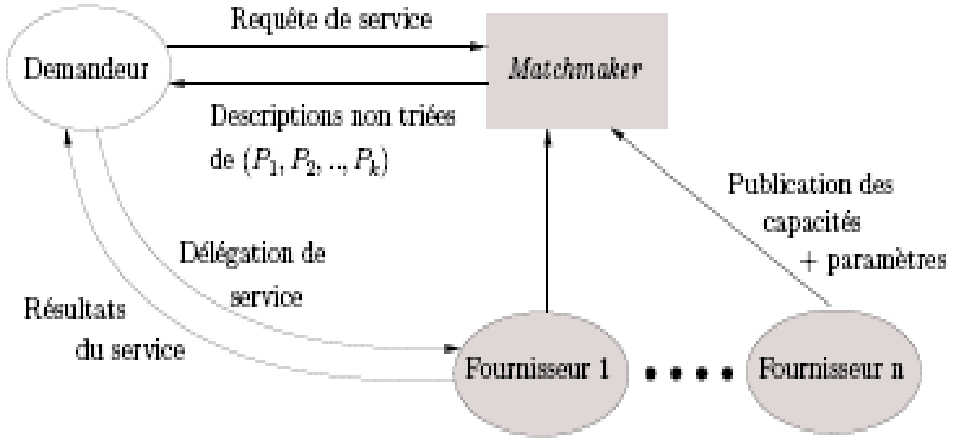
Accointances

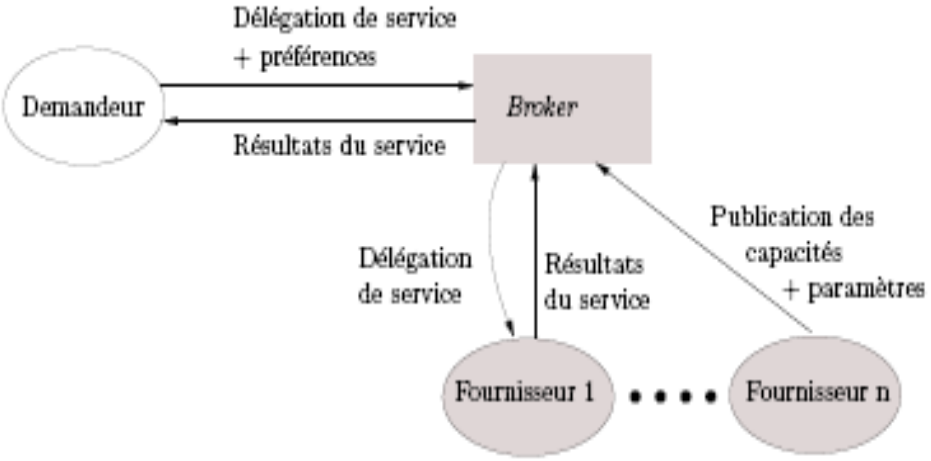
- Principe : Les connaissances sociales d'un agent représentent ses accointances avec les autres agents du système.



- Avantages
  - Simplicité,
  - Limitation a priori du coût de communication et de traitement.
- Limites
  - Gestion de l'ouverture.
  - Espace de recherche réduit.
  - Prise en compte du contexte

Middle-Agent [Sycara 00]





- Principe : stocker la connaissance sur les compétences dans des agents spécialisés.
- Modèle de coordination fondé sur les compétences (capability-based coordination)

Quoi: Théorie des actes de langage

Préambule : niveaux de communication

- Chez les êtres vivants
  - Couche physique (son, phéromone...)
  - Couche syntaxique (agencement des sons, grammaire)
  - Couche sémantique (abeilles, sens des mots)
  - Couche pragmatique (effet réel, rôle social...)
- Dans les systèmes informatiques
  - Transport (niveau « réseau »)
  - Niveau syntaxique (langages de KR)
  - Niveau connaissances (ontologie)
  - Niveau message (langage de communication)
  - Niveau communication (protocoles)

ACL

Théorie des Actes de Langage (suite)

- Austin (62), Searle (69), Vanderveken (88)

  → Parler (communiquer), c'est modifier l'état mental de ses interlocuteurs, donc c'est agir.

- Trois aspects (ou actes) d'un énoncé
  - Locutoire : l'action de dire
  - Illocutoire : l'action que souhaite l'auteur
  - Perlocutoire : ce que comprends l'interlocuteur

Th. des Actes de Langage

- Cinq catégories d'actes de langage (Searle):
  - Assertifs — faits/connaissances : pere(pierre,marie)
  - Directifs — ordres : faire(action)
  - Promissifs — engagements : ok(action)
  - Expressifs — conversation méta : bien_reçu
  - Déclaratifs — la séance est ouverte...
- Approche cognitiviste (Sperber & Wilson):

  3 actes
  - Dire que : assertions, promesses, prédictions
  - Dire de : impérative, ordres, conseils
  - Demander si : interrogative, demande d'information

Théorie AL et SMA

- Modèles fondés sur Searle, Sperber & Wilson
  - Demande d'information + Affirmation
  - Ordres + Engagements + Méta/Déclaratif
- Aspect illocutoire
  - Locutoire = support de la communication
    (couches syntaxe et connaissance)
  - Communication réussie :
    Le destinataire interprète mon message correctement
    → Perlocutoire = Illocutoire

Notion de performatif

- Contenu illocutoire
  - Force(Proposition)
    - Force illocutoire *ou* ***performatif***
    - Contenu propositionnel
  - Exemples : affirme(il_pleut), demande(il_pleut)...
- Dans les SMA
  - Performatifs issus directement des catégories d'actes de langage
    - Interrogatif + Assertif
    - Exercitif + Promissifs + Expressifs
  - Chez Ferber, *expressifs* = compétences, croyances...
  - Message = Performatif + Proposition

Interactions directes

- Message SMA :
  - Un émetteur et des destinataires
  - Un performatif + contenu propositionnel
- Gestion des messages
  - Identifiants
  - Niveau connaissance → modèle KR
- Agent Communication Language
  - Structure des messages
  - Définition des performatifs
    - Et surtout de leur sémantique...

Langage de communication agent

KQML

- Knowledge Query and Manipulation language
- Le principe est de séparer la sémantique liée au protocole de communication (indépendante du domaine de l'application) de celle liée au contenu des messages (dépendante de l'application)
- Historique :
  - 1993 du consortium DARPA-KSE (knowledge sharing effort)
  - Conçu pour échanger des informations et des connaissances entre agents
  - But : développer des outils génériques pour favoriser l'interopérabilité des agents hétérogènes

KQML

Syntaxe

Niveau message

Niveau communication

Niveau contenu

```
(KQML-performative
        :language    <texte>
        :ontology    <texte>
        :sender      <texte>
        :receiver    <texte>
        :content     <expression>
)
```

KQML
performatives

Catégorie	Nom
Demande simple	evaluate, ask-if, ask-about, ask-one, ask-all.
Demande de réponses multiples	stream-about, stream-all, eos.
Réponse	reply, sorry.
Information générique	tell, achieve, cancel, untell, unachieve,
Générateur	standby, ready, next, rest, discard, generator.
Définition de Compétences	advertise, subscribe, monitor, import, export
Gestion de réseau.	register, unregister, forward, broadcast, route

KQML

Exemple

```
(ask-one
  :sender joe
  :content (PRICE IBM ?price)
  :receiver stock-server
  :reply-with ibm-stock
  :language LPROLOG
  :ontology NYSE-TICKS)
```

KQML

Sémantique

- Pour chaque performatif
  - Préconditions (Pre) indiquent les états nécessaires dans les agents pour que le message puisse être créé.
  - Postconditions (Post) décrivent les états des agents après le traitement du message
  - Completion décrit le résultat attendu du message.
- Exemple
  - Tell(A,B,X)

Tell(A,B,X)

- Pre(A): bel(A,X) ∧ know(A,want(B,know(B,bel(A,X))∨know(B,¬bel(A,X))))
- Pre(B): intend(B, know(B,bel(A,X))∨know(B,¬bel(A,X)))
- Post(A): know(A,know(B,bel(A,X)))
- Post(B): know(B,bel(A,X))
- Completion: know(B,bel(A,X))

KQML

Discussion

- Avantages
  - Premier « standard » de communication
  - Beaucoup d'applications supportent KQML
  - Langage extensible :
    - Création de nouvelles performatives
    - Création de nouveaux paramètres
    - Création de nouvelles ontologies
- Limites
  - Des performatifs non-conformes à la théorie des actes du langage.
  - Ambiguïté et imprécision;
  - Ne prend pas en compte les conversations;
  - Classes de performatives absentes car il n'y a pas d'expressifs et de déclaratifs.

FIPA-ACL
Introduction

- FIPA = Fondation for Intelligent Physical Agents
- Elle a bénéficié des résultats de la recherche sur KQML
- Même syntaxe que KQML
- Protocoles explicites pour les échanges de messages

FIPA/ACL

- Adressage des agents
- Mécanisme de transport
- Structure d'un message
- Performatifs prédéfinis

Adressage

- Agent IDentity (AID)
  - Identifiant unique de l'agent
  - Multi-plateforme *(ex: hôte+port+numéro)*
    - → service de numérotation : AMS
- Communiquer → adresse de l'agent
  - Broadcast *(tous les agents sur une PF)*
  - Connue
    - Réponse à un message
    - Fournie par l'environnement (fermé, couplage fort)
  - Environnement ouvert:
    - Service de pages jaunes : DF

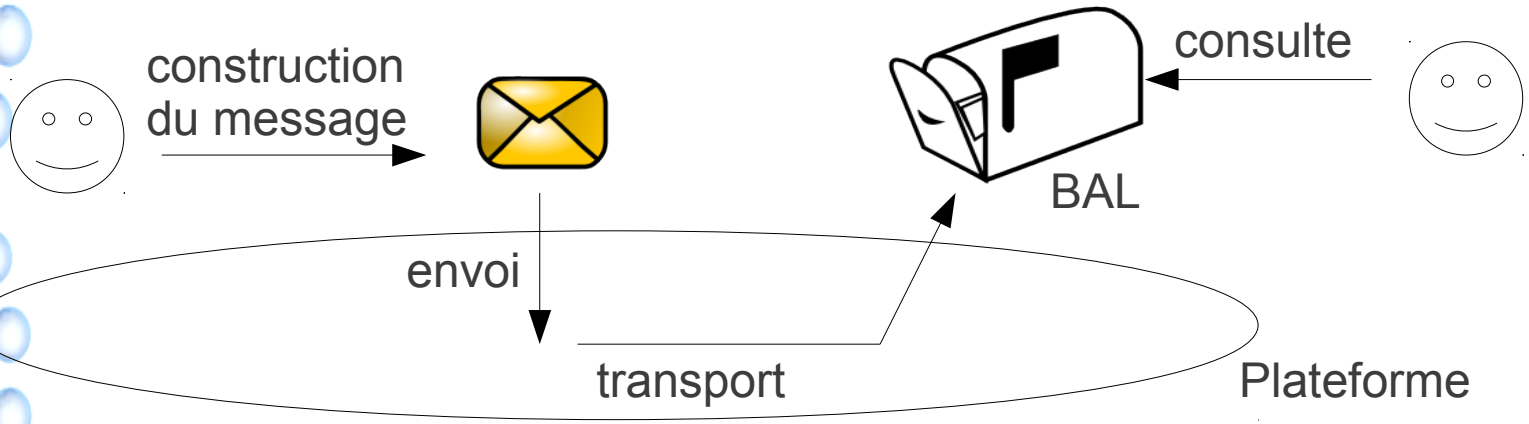
      ⇒ enregistrer (nom,AID), rechercher nom → AID

Architecture

- Transport du message

  Impossible d'utiliser de l'invocation (interblocage)

  → boites aux lettres (asynchrones)



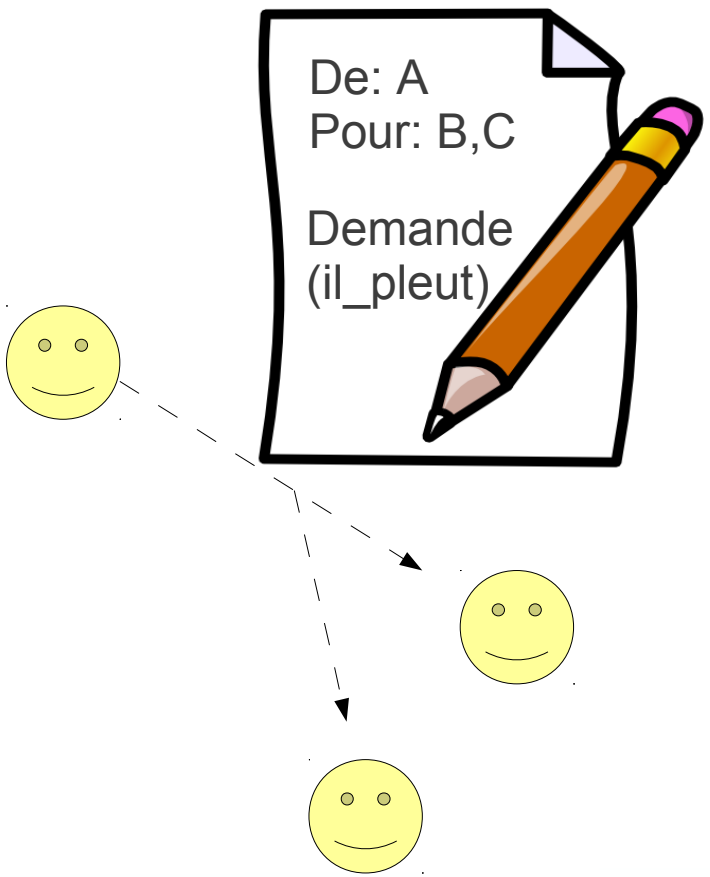
- Fonctionnement :
  - Action exogène « envoyer » asynchrone
  - 1 agent → 1 BAL – actions de consultation

    Peut être vue comme des actions exogènes
  - Filtrage des messages *(emetteur, performatif, …)*
  - Boucle procédurale : consulter + agir

    → *message = précondition*

Structure d'un message

- Enveloppe
  - Émetteur : AID
  - Destinataire : AID
  - Langage, ontologie (KR)
  - Identification du message
    - message ID
    - conversation ID
- Contenu
  - Performatif (force)
  - Contenu (proposition)



FIPA-ACL

- Représentation formelle

  <snd, performatif(dest,contenu)>

- Exemple :

  Agent *météo*, action *répondre* :

  - Précondition : ← <snd,demande(self,c)>
  - Effet :

    si (c=il_pleut ∧ temps=pluie)

    ou (c=il_fait_beau ∧ temps=soleil)

    alors → <self,oui(snd,Ø)>

    sinon → <self,non(snd,Ø)>

Reçu : ← *ou* ?

Envoyer : → *ou* !

  Agent *parisien*, action *demander* :

  - Précondition : ¬decidé ∧ ¬attente
  - Effet : → <self,demander(il_pleut,meteo)>, attente

Performatifs FIPA-ACL

22 performatifs de base

- Interrogatif
  - *Query If, Query Ref, Subscribe*
- Assertif
  - *Inform, Inform If, Inform Ref*
- Exercitif
  - *Request, Request When, Request Whenever, Propagate, Proxy*
- Promissifs
  - *Agree*
- Expressifs
  - *Cancel, Failure, Not Understood*
- Spécifiques au CNP
  - *Call for Proposal, Propose, Accept Proposal, Confirm, Disconfirm, Refuse, Reject Proposal*

Sémantique

- Modélisation BDI, système KD45
  - Modalités B, U, I → tout est « croyance »
    - B_i f : l'agent i croit que f est vrai

      Les connaissances de l'agent (locales)
    - U_i f : l'agent i croit f plus probable que ¬f

      Les propositions probables
    - I_i f : l'agent i veut que f soit vraie

      Les proposition que l'agent souhaite vraies
  - Prédicat(s) d'action : action ou <agent,action>
    - Ex : Done(<agent,action>,precond)

Performatifs FIPA-ACL (1)

- Performatif « Inform »

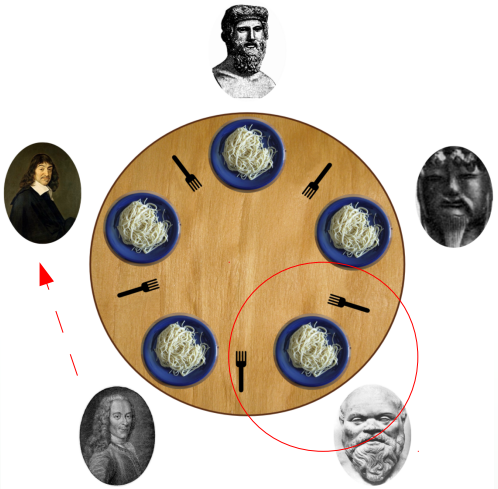
  Message $Agt_i \rightarrow Agt_j$: inform(p)

  - *i* dit à *j* que *p* est vrai
  - *i* croit que *p* est vrai (agent non-byzantin)
  - *i* ne croit pas que *j* connait déjà *p*

  Effet :

  - *j* croit que *p* est vrai

    NB: *p* peut être « négative »



- Exemple :

  <Descartes, inform(Hegels, not(holds($fourchette_2$,Platon)))>

Performatifs FIPA-ACL (2)

- Formellement : $\langle i , inform(j , p)\rangle$

  Précond: $B_i\, p \wedge \neg B_i(Bif_j\, p \vee Uif_j\, p)$

  Effet: $B_j\, p$

  avec $Bif_x\, \phi \equiv B_x\, \phi \vee B_x \neg \phi$
  et $Uif_x\, \phi \equiv U_x\, \phi \vee U_x \neg \phi$

- Performatif « Inform-if »

  Définition d'ordre syntaxique

  $\langle i , informIf(j , p)\rangle \equiv \langle i , inform(j , p)\rangle \vee \langle i , inform(j , \neg p)\rangle$

  Précond: $B_i\, p \wedge \neg B_i(Bif_j\, p \vee Uif_j\, p)$

  Effet: $B_j\, p$

  - *i* dit à *j* si *p* est vrai ou faux
  - *i* connait la valeur de vérité de *p* (ou de *!p*)
  - *i* ne croit pas que *j* la connait

Performatifs FIPA-ACL (3)

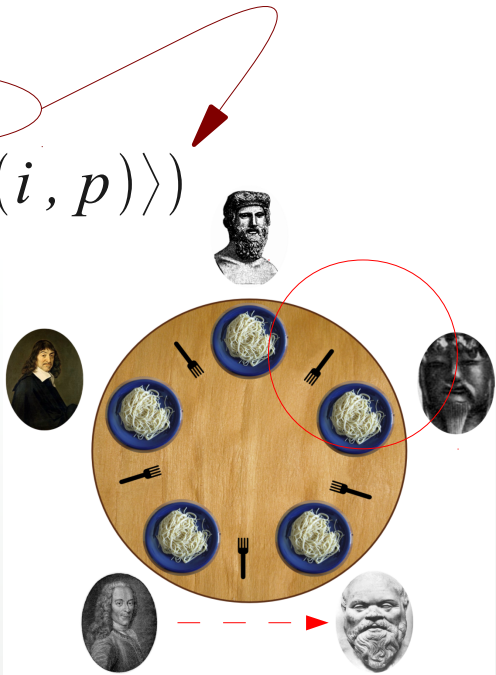
- Performatif « Query-if »

  Message $Agt_i \rightarrow Agt_j$: query-if(p)

  - i demande à j si p est vrai
  - i ne connait pas la valeur de vérité de p

    → i ne croit pas p et i ne croit pas !p
  - i croit que j peut lui donner cette valeur

$$\langle i, queryIf(j, p)\rangle \begin{cases} \text{Précond: } \neg Bif_i\, p \wedge \neg Uif_i\, p \wedge B_i\, Bif_j\, p \\ \qquad\qquad \wedge \neg B_i\, I_j\, Done(\langle j, informIf(i, p)\rangle) \\ \text{Effet: } Done(\langle j, informIf(i, p)\rangle) \end{cases}$$



- Exemple :

  <Descartes, query-if(Hegels, holds(fourchette$_2$,Platon))>

Performatifs FIPA-ACL (4)

- Performatif « Request »

  Message $Agt_i \rightarrow Agt_j$: request(p)

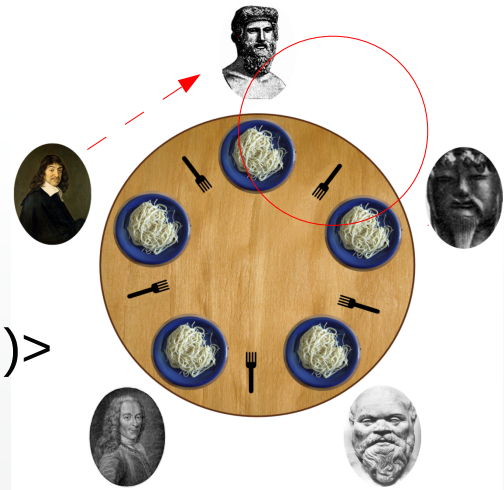
  - i demande à j de faire X
  - i croit que <i,X> est possible

    i.e. l'ensemble des préconditions est satisfaite
  - i croit que j peut faire X
  - i croit que j n'a pas déjà l'intention de faire X

  Effet : X a été effectuée

- Exemple :

  <Descartes, request(Platon, lacher(fourchette$_2$,Platon)))>



Performatifs FIPA-ACL (5)

- Formellement

$$\langle i\,,\,request\,(\,j\,,\,a\,)\rangle$$

Précond: $P\,(a)_{[i/\,j]} \wedge B_i\,Agent\,(\,j\,,\,a\,) \wedge \neg\, B_i\,I_{\,j}\,Done\,(a\,)$

Effet: $Done\,(a\,)$

avec $P\,(a\,)$ = préconditions de a
et $P\,(a\,)_{[i/\,j]} = \{\,p \in P\,(a\,)$ tq $B_i\,p\}$

- Remarque : *query-if = request*

$$\langle i\,,\,queryIf\,(\,j\,,\,p\,)\rangle \equiv \langle i\,,\,request\,(\,j\,,\langle\,j\,,\,InformIf\,(\,i\,,\,p\,)\rangle\,)\rangle$$

→ i demande à j de lui donner la valeur de p

Préconditions induites !

- i croit j peut le faire

  → donc B_i Bif_j p *(d'après les préconds de inform)*

- i croit que j n'a pas déjà l'intention de le faire

Remarques

- Performatif *request* :

  L'agent i ne peut pas faire a $\neg B_i$ Agent(i,a)

  ou il n'en a pas l'intention $\neg D_i$ Done(<i,a>)

  ou il est incapable d'atteindre les préconditions

  → Pas définies dans FIPA !

- Réponse à *request* : → protocoles

  agree, *refuse*, *inform-done*, …

Performatifs FIPA-ACL (6)

- « Quelle heure est-il ? »

  → Impossible avec un Query-if

- Query-ref

  <i, query-ref(j, Ref x δ (x))>

  δ(x) = expression en x

  Ref = ι, all, any

  - Iota = description des propriétés
  - All = tous les x possibles
  - Any = n'importe lequel

→ Donne moi le x qui vérifie cette propriété

Performatifs FIPA-ACL (7)

- Exemples

  <i, query-ref(j, all x (pull-violet x))>

  → Dis-moi qui sont les étudiants qui portent un pull violet

  <i, query-ref(j, any x (pull-violet x))>

  → Donne-moi n'importe quel étudiant qui porte un pull violet

  <i, query-ref(j, iota x (pull-violet x))>

  → Qui est cet étudiant qui porte un pull violet ?

- Inform-ref

  <i, inform-ref(j, Ref x δ(x) = r)> r est la « réponse »

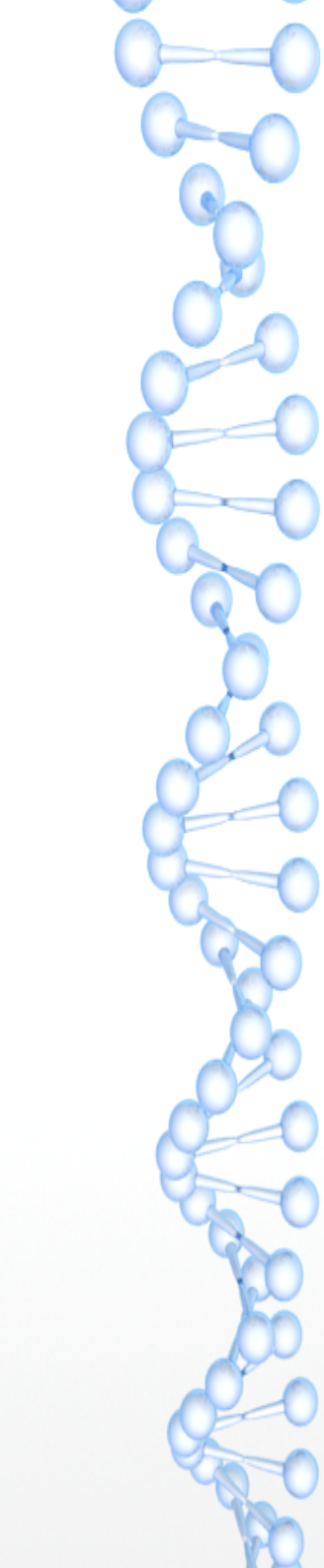
  Ex : <i, inform-ref(j, any x (pull-violet x) = bill)>

Différences entre ACL et KQML

- Syntaxe identique sauf pour les primitives
  (ex : « tell » pour KQML, « inform » pour ACL)
- différence dans les descriptions sémantiques:
  - Définition des primitives différentes
    - **pré-conditions et les post-conditions pour KQML**
    - **Faisabilité et effet pour ACL**
  - Utilise un langage différent pour décrire les états mentaux des agents
    - Mais les deux considèrent des agents coopératifs (locutoire = illocutoi

ACL est un standard de la FIPA :

- Évolution plus lente que KQML
- ACL est plus portable (supporté par quelques plateformes, e.g. JADE)



Autonomie d'interaction

- Interactions directes

  → Non-prédictibilité de la réponse

  La sémantique de l'interaction ne décrit pas la réponse, seulement les conditions et effets du point de vue de l'agent émetteur !

- L'agent peut répondre ce qu'il veut

  → définition de protocoles

  en particulier, SMA faiblement couplés

- L'agent peut ne pas répondre

  → attente et timeout

  en particulier, SMA ouverts

Protocole

Introduction

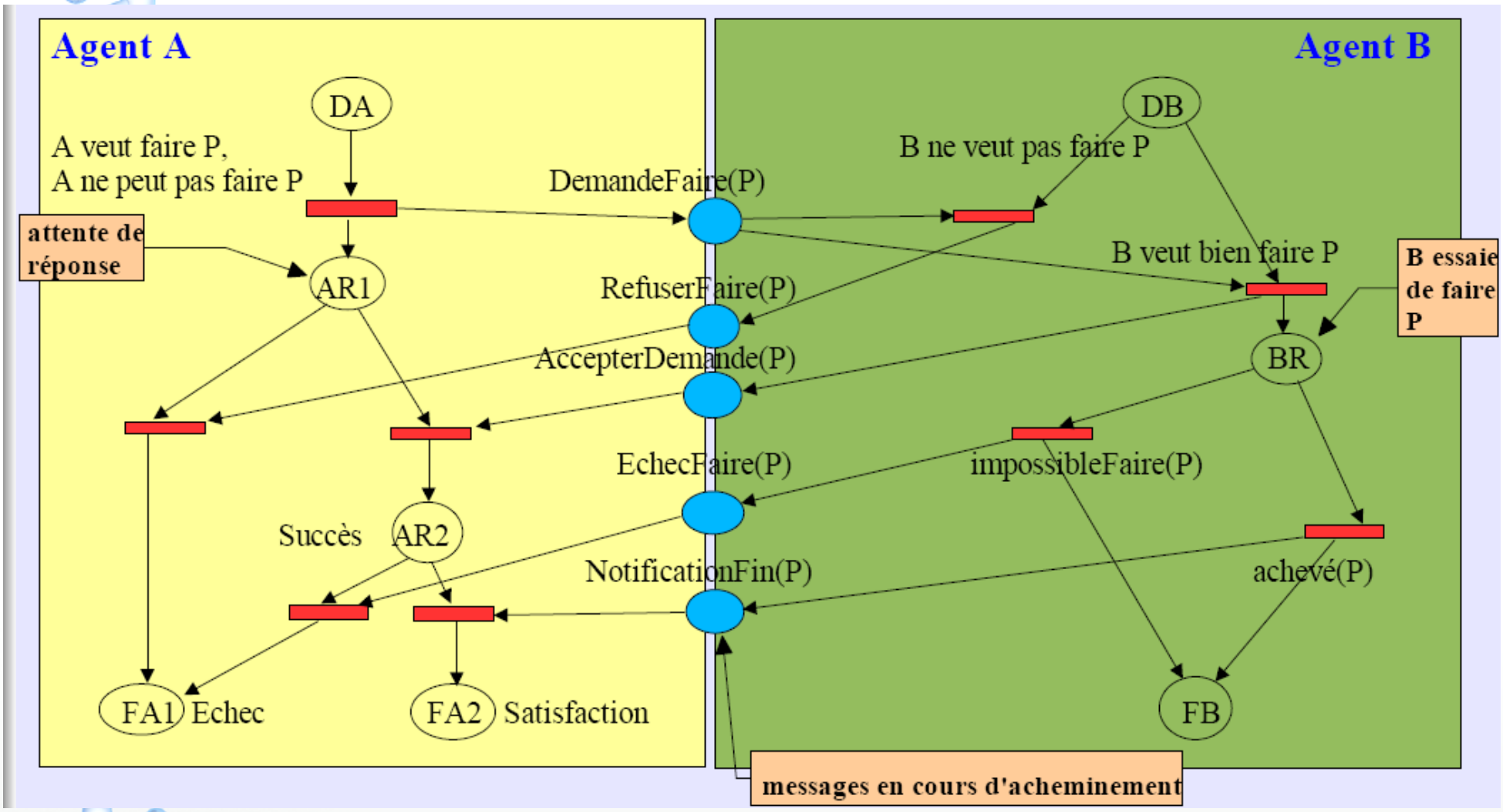
- Problématique :
  - intégration de schémas typiques structurant les échanges.
  - Permettre aux agents de savoir comment utiliser les ACL.
- Définition
  - Un protocole précise qui peut dire quoi à qui et les réactions possibles à ce qui est dit.
- Conséquences
  - Une restriction de l'utilisation des actes.
  - Ingénierie des protocoles
    - Protocoles prédéfinies
    - Formalisme de définition des protocoles
    - Méthodologie de définition de protocoles.

Protocole formalisme
Réseau de Pétri

- Une conversation est une suite d'états liés par des transitions.
- Places
  - État interne de l'agent
  - Message en cours d'acheminement
- Transitions
  - Synchronisation due à la réception de messages,
  - Conditions d'application des actions.

Réseau de Pétri

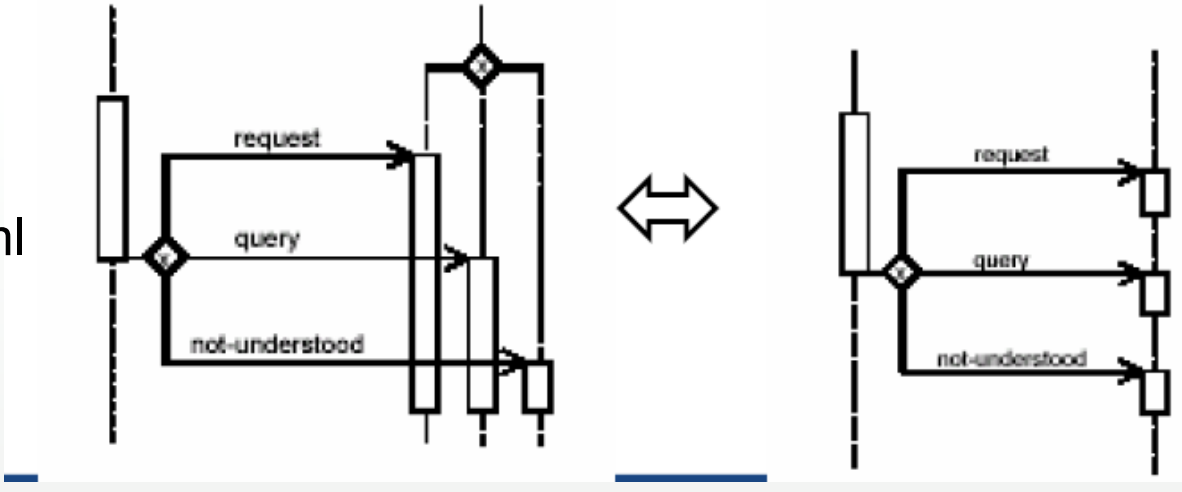
Exemple



Interaction Formalisme AUML

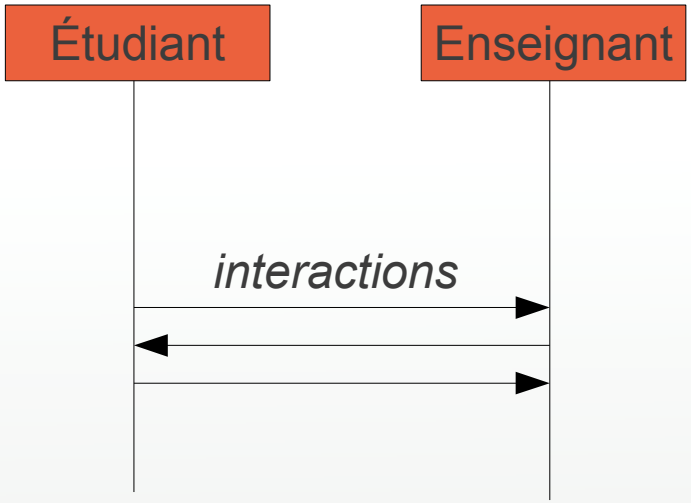
- Sémantique : un protocole est un ensemble de messages échangés entre des rôles
- Notation
  - Une dimension verticale qui représente le temps.
  - Une dimension horizontale qui représente les différents rôles.

www.auml.org/auml/projects/main.shtml



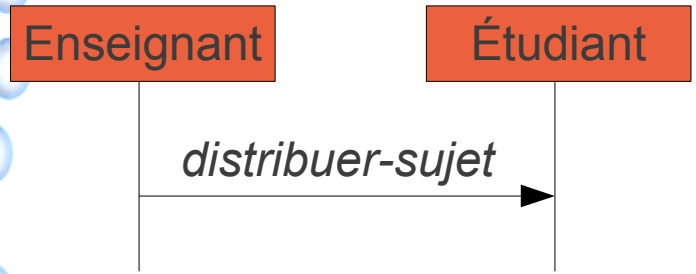
Protocole : rôle

- Exemple :
  - Rôles : étudiant, enseignant
  - Agents : Anne, Bob, Charles
  - Anne → étudiant, Bob → étudiant, enseignant
- Fil d'interaction
  - → à partir des rôles

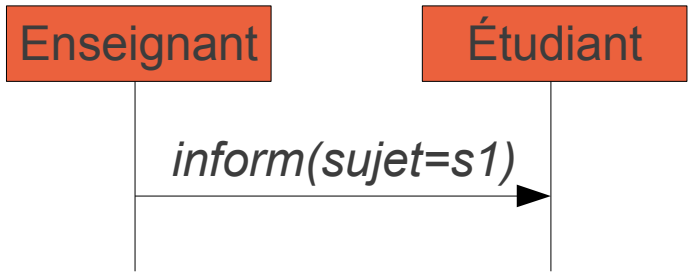


Protocole : message

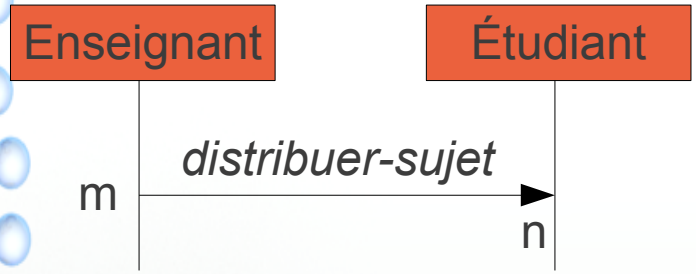
- Performatif



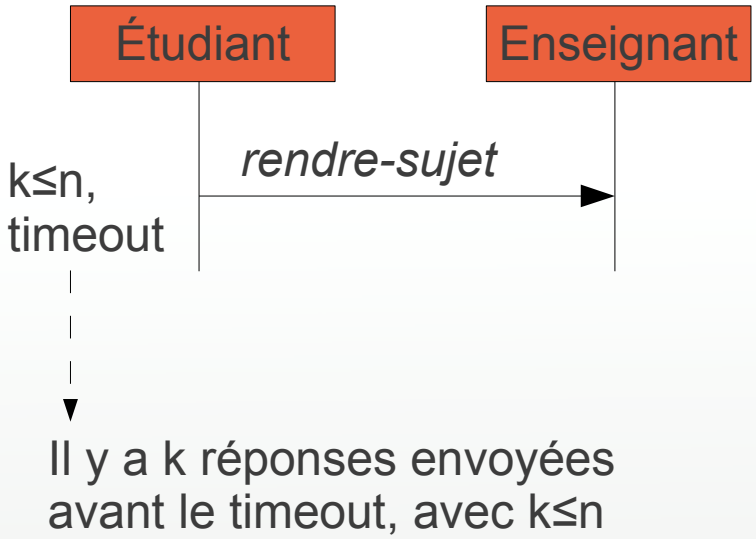
- Performatif + Contenu



- Arité

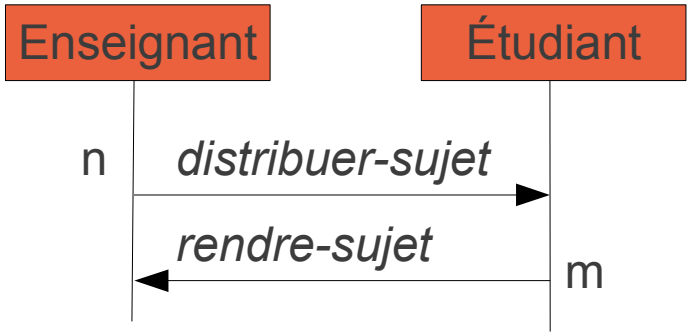


- Condition



Protocole : messages

- Agencements



→ si enseignant envoie un « distribuer-sujet » alors il recevra un « rendre-sujet »

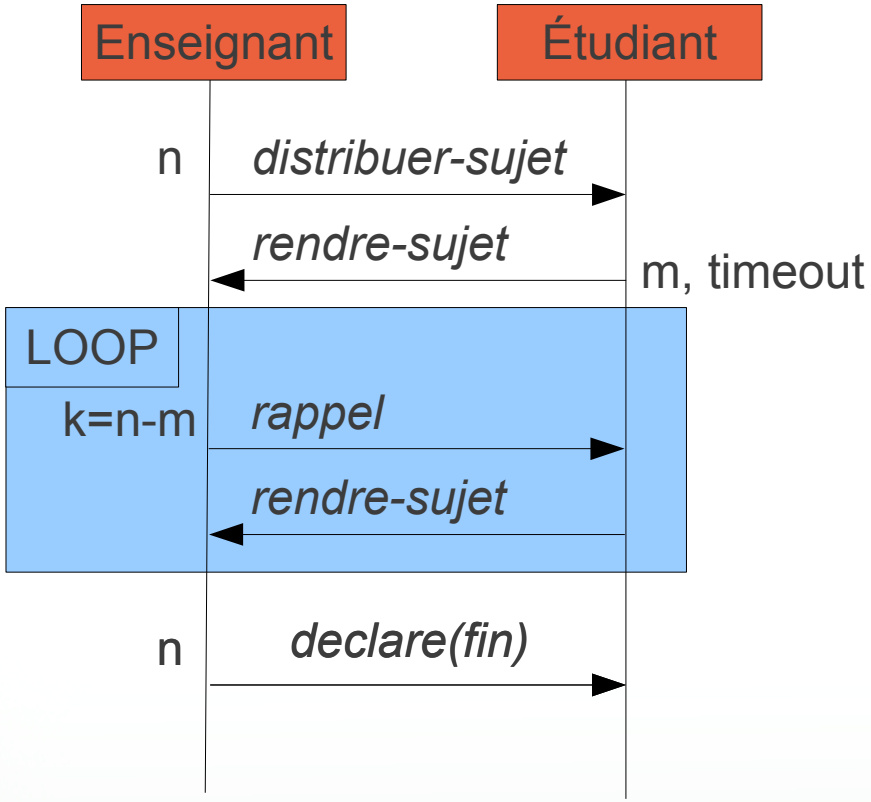
- Attention (rappel)

Ce sont des rôles, pas des agents !

→ Les agents qui prennent un rôle s'engagent à interagir conformément au protocole

Protocoles : blocs LOOP

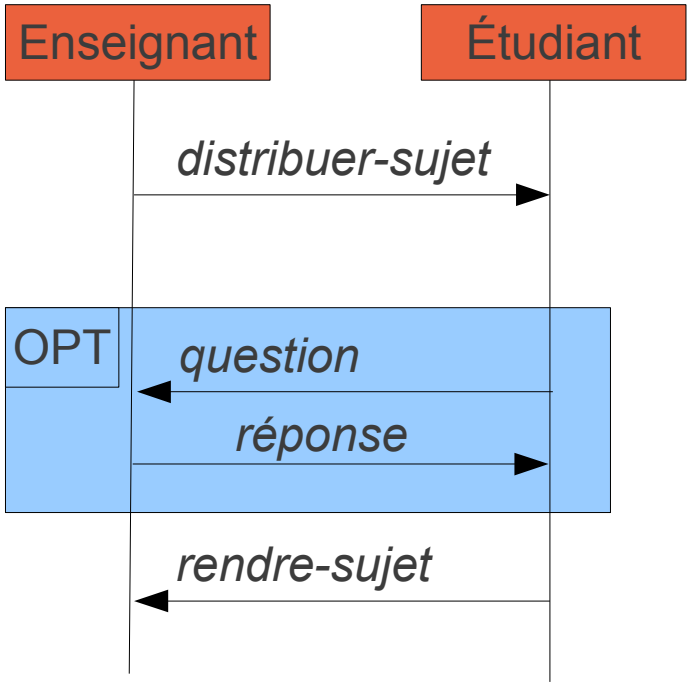
- Répéter une interaction



Après un message *rendre-sujet*, on peut avoir une infinité de *rappels*, suivis de *rendre-sujet*

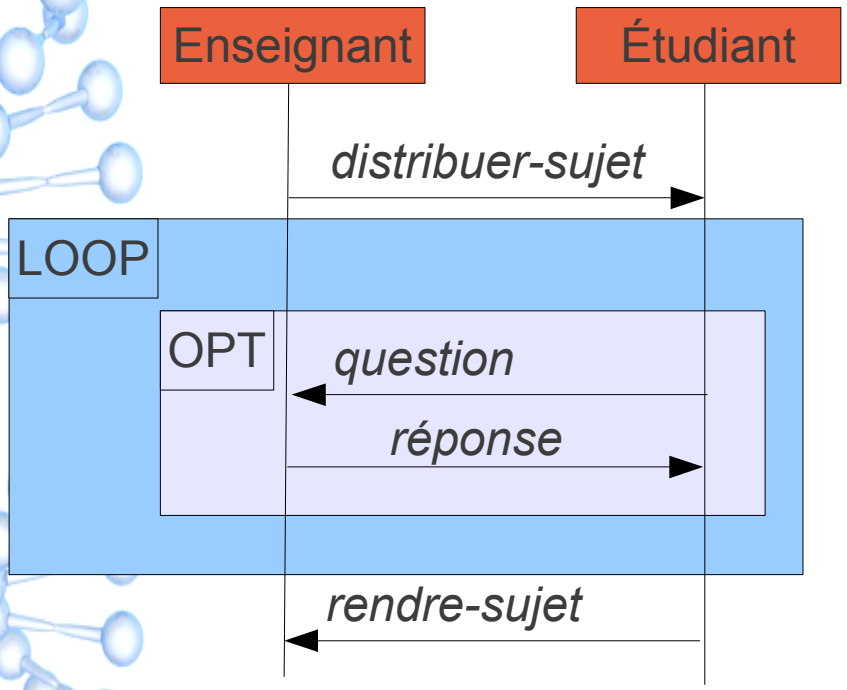
Protocoles : blocs OPT

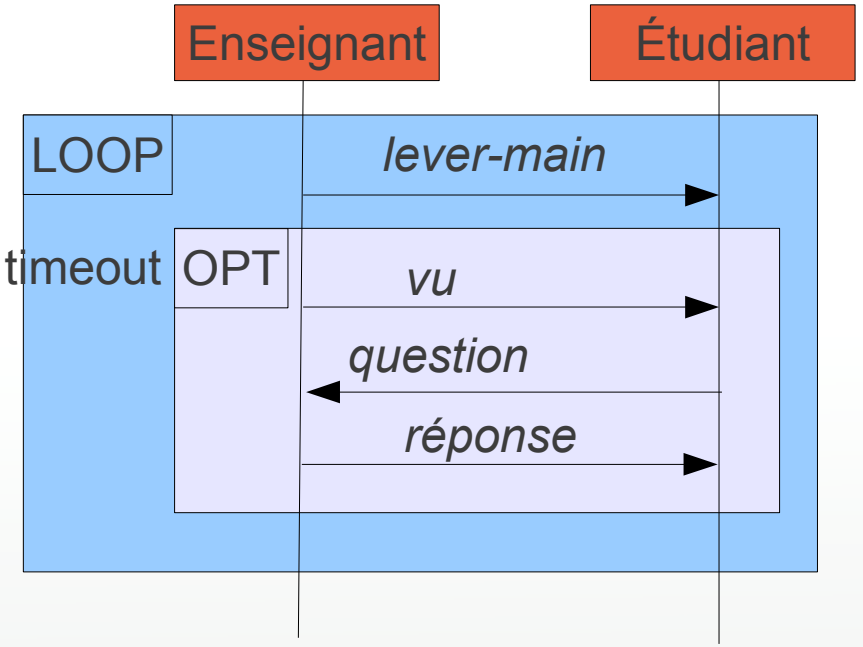
- Messages optionnels



→ sur cet exemple, il y a au plus un seul échange question/réponse

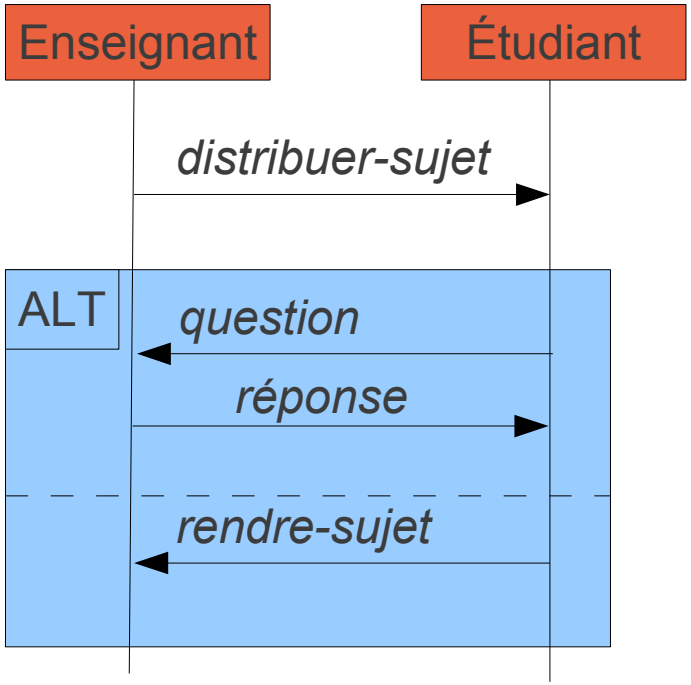
Protocoles : blocs imbriqués





Protocoles : blocs ALT

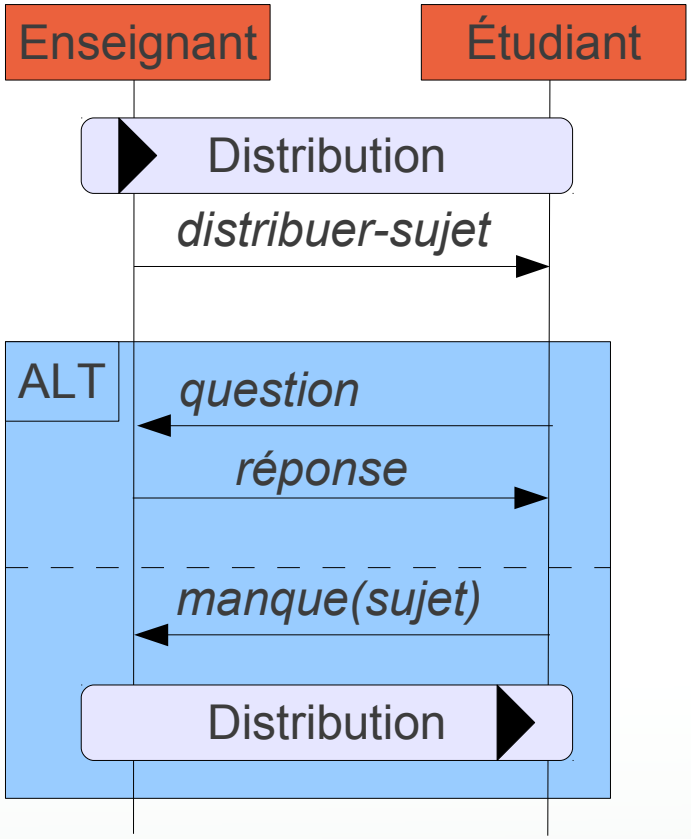
- Alternatives



→ Enseignant distribue le sujet puis soit les étudiants posent une question, soit ils rendent le sujet

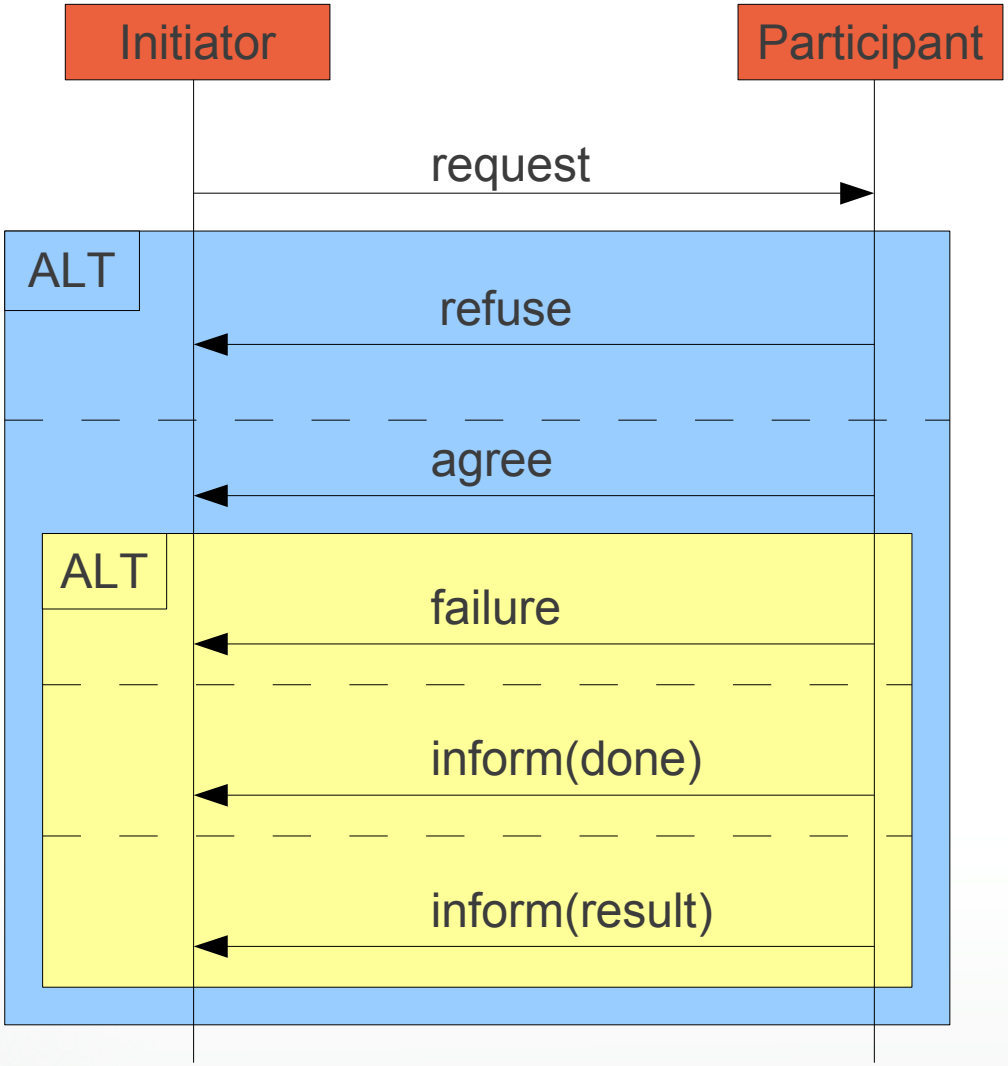
Protocoles : étiquettes

- « Macros » dans le protocole



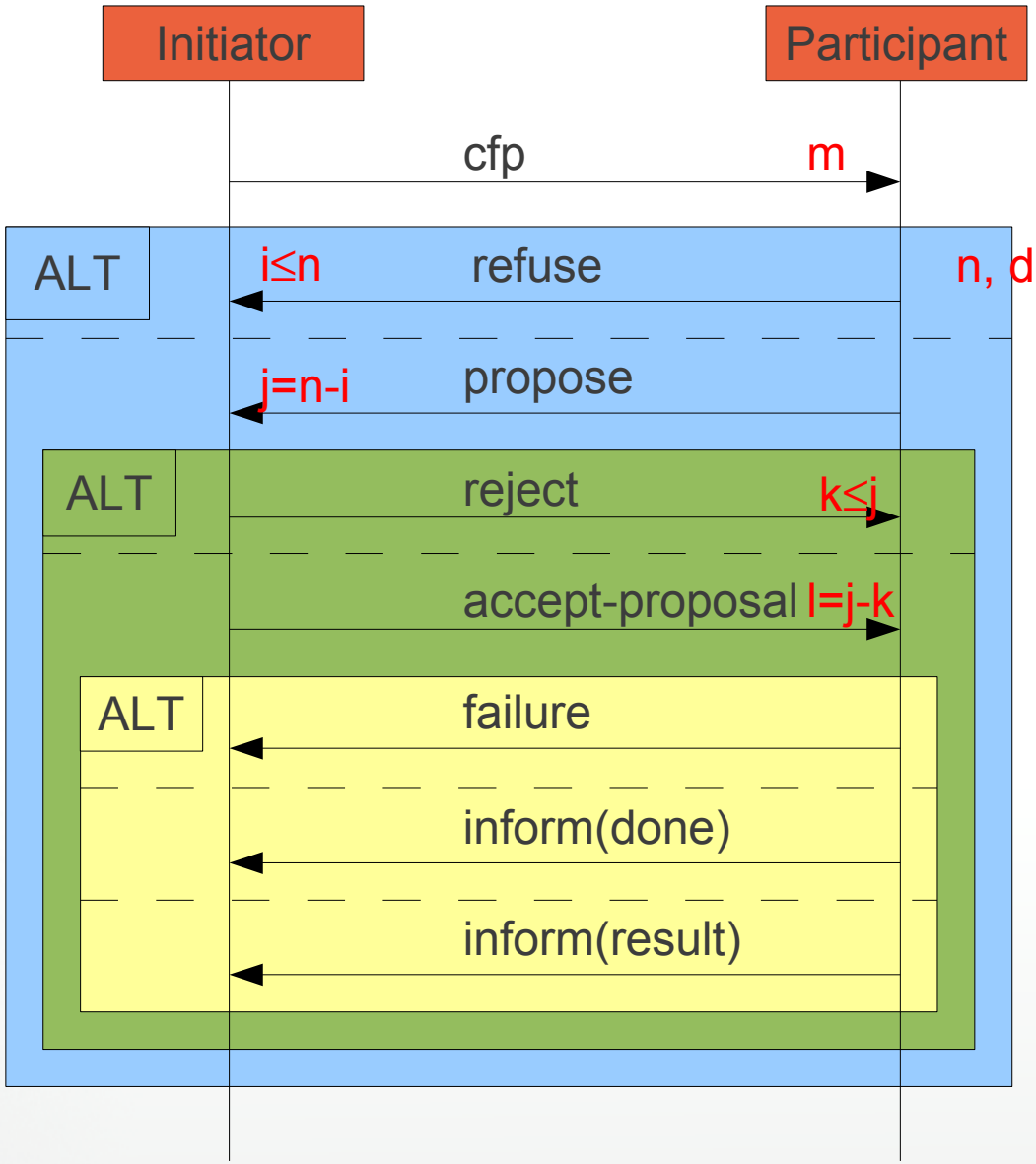
Protocole FIPA-request

FIPA N°26, 2002



Le protocole Contract Net (CNP)

FIPA N°29, 2002



m = nbr de participants
n = nombre de réponses
i = nombre de refuse
j = nombre de propose
k = nombre de reject
l = nombre de accept
deadline = date limite de réponse

Protocoles

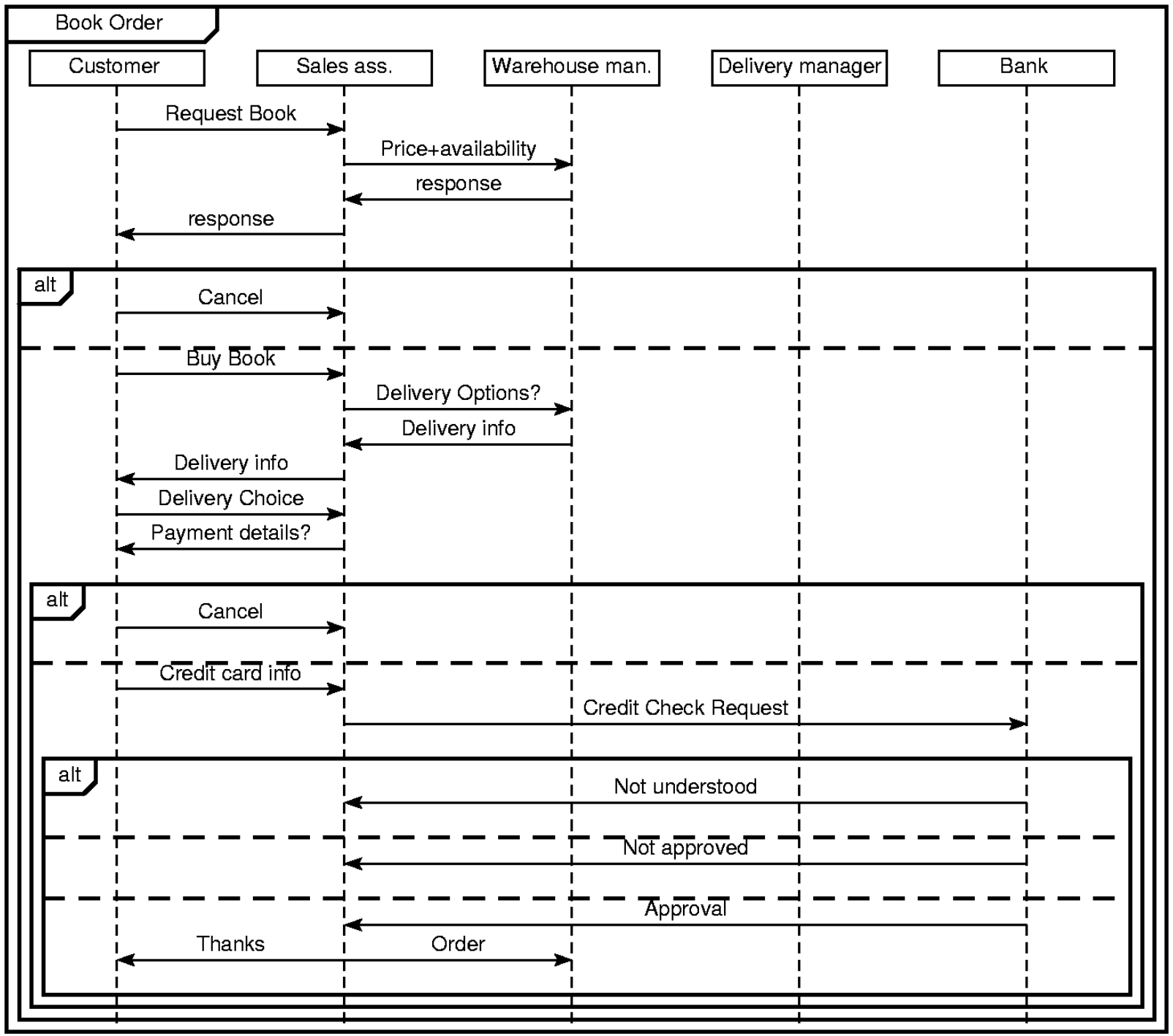
- Remarques
  - Un rôle peut être pris par plusieurs agents
  - Un agent peut prendre plusieurs rôles
- Deadlines
  - → Locales (temps propre)

    Ex : CNP → participant évalue temps de réponse et respect de la deadline
- Vision « locale » d'un problème
  - → 1 SMA = plusieurs protocoles

  Ex: annuler un *request*

  cancel (avec réponse inform ou failure)

  → pas décrit dans FIPA-request !


Book Order
Customer
Sales ass.
Warehouse man.
Delivery manager
Bank
Request Book
Price+availability
response
response
alt
Cancel
Buy Book
Delivery Options?
Delivery info
Delivery info
Delivery Choice
Payment details?
alt
Cancel
Credit card info
Credit Check Request
alt
Not understood
Not approved
Approval
Thanks
Order